IN THE SPECIFICATION:

Please replace paragraph [0001] with the following amended paragraph:

[0001] The present invention is related to the commonly owned, co-pending U.S. patent application 10/083,075, entitled "Application Portability And Extensibility Through Database Schema And Query Abstraction," filed February 26, 2002, now U.S. Patent No. 6,996,558 issued February 7, 2006; and commonly owned co-pending application, entitled "Dynamic Functional Module Availability," filed on November 24, 2003, Application Serial No. 10/720,960 ~~herewith~~ (Attorney Docket No. ROC920030277US1), which are herein incorporated by reference.

Please replace paragraph [0035] with the following amended paragraph:

[0035] Illustratively, the memory 112 contains an operating system 124. Examples of suitable operating systems, which may be used to advantage, include distributions of the Linux® operating system and versions of the Microsoft[['s]] Windows® operating system, as well as any operating systems designed for handheld devices, such as Palm OS®, Windows® CE, and the like. More generally, any operating system supporting the functions disclosed herein may be used.

Please replace paragraph [0044] with the following amended paragraph:

[0044] For some embodiments, the query 220 may be an abstract query including a set of one or more query conditions and a specified results set, each based on logical fields defined in the DRA component 148 (shown in FIG. 1). As previously described, abstract queries may be executed by the query execution component 150. In the exemplary abstract data model, the logical fields are defined independently of the underlying data representation being used in the DBMS 154, thereby allowing queries to be formed that are loosely coupled to the underlying data representation 214. The query execution component 150 is generally configured to transform abstract queries into concrete queries compatible with the physical data representation (e.g., an XML representation 214_1, an SQL representation 214_2, or any other type of representation

214_3), by mapping the logical fields of the abstract queries to the corresponding physical fields of the physical data representation 214. The mapping of abstract queries to concrete queries, by the query execution component 150, is described in detail in the commonly owned, co-pending U.S. patent application 10/083,075, entitled "Application Portability And Extensibility Through Database Schema And Query Abstraction," filed February 26, 2002, now U.S. Patent No. 6,996,558 issued on February 7, 2006.

Please replace paragraph [0046] with the following amended paragraph:

[0046] The application 120 may pass the multi-analysis plug-in 161 the newly acquired result set 222 along with other required input parameters. Each analysis plug-in 162 invoked by the multi-analysis plug-in 161, may utilize a generic interface or signature as described in commonly owned co-pending application, entitled "Dynamic Functional Module Availability," filed on November 24, 2003, Application Serial No. 10/720,960 ~~herewith~~ (Attorney Docket No. ROC920030277US1). Further, each plug-in can accept a result set data object 165 as an input parameter and produce a result set data object 165 as an output parameter. For example, a query relating to all the micro-array data for a given experiment, is built and submitted to the query execution runtime 150, via the query building interface 122. Further, the user desires that two subsequent operations be performed on the result set 222 returned by the query execution runtime 150. First, the micro-array contained in the result set 222 needs to be normalized using a normalization plug-in, P1. Once the array is normalized, the next operation will call another plug-in, P2, to rank the normalized genes. After processing is completed, P1 will produce a result set data object 165 for use as an input parameter for P2. P2 will also produce a result set data object 165 after its processing is completed.